

University of Groningen

Fuzzy Clustering, Genetic Algorithms and Neuro-Fuzzy Methods Compared for Hybrid Fuzzy-First Principles Modeling

Lith, Pascal F. van; Betlem, Ben H.L.; Roffel, Brian

Published in:
Systems Analysis Modelling Simulation

DOI:
[10.1080/02329290290031350](https://doi.org/10.1080/02329290290031350)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2002

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Lith, P. F. V., Betlem, B. H. L., & Roffel, B. (2002). Fuzzy Clustering, Genetic Algorithms and Neuro-Fuzzy Methods Compared for Hybrid Fuzzy-First Principles Modeling. *Systems Analysis Modelling Simulation*, 42(4), 597-630. <https://doi.org/10.1080/02329290290031350>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

FUZZY CLUSTERING, GENETIC ALGORITHMS AND NEURO-FUZZY METHODS COMPARED FOR HYBRID FUZZY-FIRST PRINCIPLES MODELING

PASCAL F. VAN LITH*, BEN H.L. BETLEM
and BRIAN ROFFEL

*Research group Dynamics and Control of Processes, Department of
Chemical Technology, University of Twente, P.O. Box 217,
7500 AE Enschede, The Netherlands*

(Received 21 August 2000)

Hybrid fuzzy-first principles models can be a good alternative if a complete physical model is difficult to derive. These hybrid models consist of a framework of dynamic mass and energy balances, supplemented by fuzzy submodels describing additional equations, such as mass transformation and transfer rates. Identification of these fuzzy submodels is one of the main issues in constructing hybrid models. In this paper, a new approach to constructing hybrid fuzzy-first principles models is presented, which uses a Kalman filter for parameter estimation. In addition, a comparison between three classes of identification techniques for fuzzy submodels is presented: fuzzy clustering, genetic algorithms and neuro-fuzzy methods. The comparison is illustrated for a penicillin fed batch-reactor test case. Fuzzy clustering proved to be the most suitable technique, with genetic algorithms being a good alternative.

Keywords: Hybrid models; Fuzzy logic; Clustering; Genetic algorithms; Neuro-fuzzy methods; Kalman filtering

1. INTRODUCTION

For controller design as well as prediction of process behavior, good input–output representation of a process is of paramount importance.

*Corresponding author. E-mail: p.f.vanlith@ct.utwente.nl

In addition, it is required that these models are interpretable, in the sense that, by analyzing the model, there is a physical understanding of the process behavior. Many process models are available in a state-space representation. A *framework* of mass, component and energy balances describing the essential process accumulation can often be derived from given requirements. Within this framework, effects such as reaction rates or mass transfer are described by static empirical relations. However, for many processes, these empirical relations are complex and may have limited validity and, thus, introduce uncertainty. Examples of such processes are biotechnological processes and polymerization processes, but in principle, uncertainty can be present in any process model.

Fuzzy logic could be used to deal with this uncertainty. This technique is capable of describing highly non-linear relations in a fairly simple way, without the loss of interpretability that other black box techniques have. This is a major advantage. In addition, fuzzy logic is capable of capturing human experience in a transparent way, so that these can be used to improve the process models in areas where little or no physical data exists.

By combining fuzzy logic submodels with a physical model framework, *hybrid fuzzy-first principles models* are obtained. Combining black box techniques (such as neural networks) with physical equations is not new. However, until now, little research has been presented in which fuzzy logic is used. This paper will argue that, certainly with respect to interpretability and transparency, fuzzy logic is an extremely suitable technique to be used in hybrid modeling.

To construct hybrid fuzzy-first principles models, appropriate fuzzy submodels need to be identified from human experience or process data. A problem occurs when no process measurements or human experience is available about the parameter or relationship that is to be described by a fuzzy submodel. This situation is not unlikely to appear in a real-life industrial environment. In this case, parameter or state estimation techniques can be used to solve the problem. The estimations can then be supplied to a fuzzy model identification technique to construct the fuzzy submodel.

This paper will present a procedure to construct hybrid fuzzy-first principles models from process data, by making use of a Kalman filter for parameter estimation. Furthermore, three different classes

of identification techniques for fuzzy models will be compared: fuzzy clustering, neuro-fuzzy methods and genetic algorithms.

2. HYBRID MODELING

Hybrid modeling (also denoted as semi-parametric, semi-mechanistic or polytopic modeling by some researchers) in this context denotes the combination of a black box modeling technique, in this case fuzzy logic, with first principles descriptions. Some research has been done in this area [1–4], though little of this research focuses on the use of fuzzy logic in hybrid models. Generally speaking, a distinction is made between a *modular approach* and a *semiparametric approach*, which itself falls apart into a serial and a parallel approach [1].

In modular design approaches, several blocks of fuzzy logic submodels are combined to constitute the process model. The structure of the overall model is determined using prior knowledge, while every block calculates one specific variable or parameter. Advantages of this approach are that it may improve interpretability (when compared to an overall black box model) and that it can reduce the number of model parameters. A major disadvantage is that good output behavior is not guaranteed because the combination of the blocks could generate an overall divergent behavior [1].

With semiparametric modeling, a fuzzy logic submodel is placed in tandem with a physical model. The physical model structure is fixed and derived from first principles. In the serial approach, the fuzzy logic submodel calculates intermediate variables to be used in the parametric model (Fig. 1). In this case, divergence is not likely to

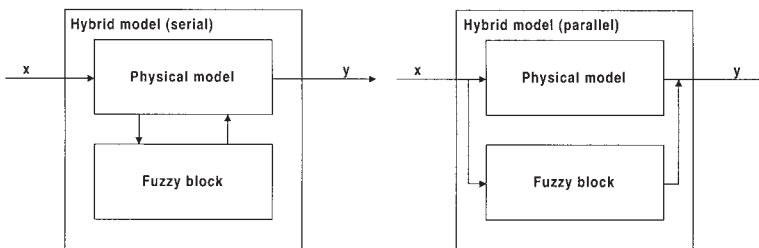


FIGURE 1 Semiparametric hybrid modeling.

occur, although this depends on the quality of the fuzzy logic submodel. In the parallel approach, the outputs of the fuzzy logic block and the physical model are combined to determine the total model output (Fig. 1). The model serves as a best estimate of the process. The fuzzy logic submodel is implemented such, that it is able to compensate for any discrepancy between the prior model output and real world values. A disadvantage of this approach is that desired behavior is not guaranteed.

Research in this area has, until now, focused on Artificial Neural Networks (ANNs) as the black box modeling technique. In the context of modeling chemical processes, however, it is proposed to use fuzzy logic, as mentioned above, because fuzzy logic is capable of capturing data as well as human experience, can deal with uncertainty and can usually be interpreted physically.

If first principles models are preferred over black box models, it is proposed to leave the physical model structure intact as far as possible and only model those phenomena about which uncertainty exists with fuzzy submodels. The physical model structure is formed by mass and energy balances, while the fuzzy submodel(s) describe production rates, heat and mass transfer, equilibria or growth rates, or any additional equation, if required. This way, hybrid fuzzy-first principles are obtained which combine a high level of interpretability with the expectation of good extrapolating properties. Therefore, in this research, a serial semiparametric modeling approach is used.

3. PROCEDURE

The procedure for constructing a hybrid model *from process data* is straightforward and can be formulated as:

1. Establish the physical framework of the hybrid model, by determining for which relations uncertainty is present, while leaving the accumulation balance structure intact.
2. Acquire input–output process data about uncertain relations (by estimation, if required).
3. Identify the fuzzy submodel(s) for these uncertain relations, using the acquired input–output data.

4. Integrate the fuzzy submodel(s) into the established accumulation balances to obtain the hybrid model.

The remainder of this paper will discuss these steps with respect to a selected test case in more detail.

4. TEST CASE

To illustrate the hybrid modeling procedure, a biotechnological modeling problem has been selected. It is well known that it is difficult to develop a first principles description for such processes. The process under study is a penicillin fermentation fed-batch reactor, for which a physical model was available [1]. This model is used as a basis for developing a hybrid model and is used as a reference in the comparison of the various hybrid models that will be developed. Obviously, using this approach it is not possible to determine whether the hybrid models will outperform the known physical model. However, it is possible to illustrate the modeling procedure and to compare the different modeling techniques and it is an approach often used by other researchers, amongst others, [5].

The model consists of four state equations, describing biomass concentration X (gDCW/l), substrate concentration S (g/l), product concentration P (g/l) and the overall reactor volume V (l):

$$\frac{dX}{dt} = X(\mu - D - c_L) \quad (1)$$

$$\frac{dS}{dt} = -\sigma X + (S_f - S)D \quad (2)$$

$$\frac{dP}{dt} = q_p X - P(D + K) \quad (3)$$

$$\frac{dV}{dt} = F \quad (4)$$

where μ is the specific growth rate (h^{-1}), c_L is the cell lysis rate (h^{-1}), D is the dilution rate (h^{-1}), σ is the substrate consumption rate (h^{-1}), S_f is the substrate concentration in the feed (g/l), q_p is the product

TABLE I Settings input and process noise

<i>Variable</i>	<i>White noise amplitude</i>
<i>F</i>	$[-0.005, 0.005]$ l/h
<i>S</i>	$[-0.2, 0.2]$ g/l
<i>X</i>	$[-0.1, 0.1]$ gDCW/l

formation rate (h^{-1}), K is the product decay constant (h^{-1}) and F is the feed rate (l/h). A complete description of the model is given in Appendix A. While generating process data using this model, noise was added: the settings are given in Table I. For the simulations, a discrete version of the model above (Eq. ((1)–(4))) is implemented. White noise of a certain bandwidth is added to the reference model process state at time step k . Since the state of time step k is used to calculate the state at time step $k + 1$, some correlation occurs.

5. Establishing the Hybrid Model Structure

To test the hybrid modeling procedure, it was assumed that the rate equations in the state equation for the biomass concentration, describing the specific growth rate and the cell lysis rate, were inaccurate. The two rates were lumped together to form a net growth rate α . Since the specific growth rate and the cell lysis rate both are functions of S and X , α will also be a function of S and X . The goal is then *to develop a hybrid model for the fed-batch reactor in which α is governed by a fuzzy submodel*. Equation (1) then has to be replaced by the following equation:

$$\frac{dX}{dt} = X(\alpha - D) \tag{5}$$

where α is the net growth rate (h^{-1}). Furthermore, it is assumed that α is not measurable, while the states of the model (S , X , P and V) can be measured. Since the fuzzy submodel will be identified using input–output data, the Sugeno (TSK) type fuzzy model is selected. These models are usually less complex than linguistic fuzzy models (fewer rules), which helps transparency.

6. ACQUIRING INPUT-OUTPUT DATA

The need for input-output data of the uncertain relation that is to be described by a fuzzy submodel is in this case only imposed by the cluster algorithm. Since the genetic algorithm approach and neuro fuzzy methods are based on “training” of fuzzy models, they can be implemented in such a way that they can be used to “train” a complete hybrid model (that is, training a fuzzy submodel within a hybrid model). This omits the need for actual input-output data about the uncertain relation, because the output of the hybrid model is used, which can be compared with the appropriate process measurements. However, to make a fair comparison, the techniques presented here are compared for identifying fuzzy submodels using input-output data of the uncertain relation.

Acquiring input-output data for this problem is less complicated than for industrial plants. For the latter type of plants, two items need particular attention. First of all, the majority of the process data is available for the process under normal (usually constant) conditions, while process data about variations or upsets is limited. However, information about variations is extremely valuable for modeling purposes. Secondly, the process data that is available is often that of the *controlled* process. Thus it does not represent the open-loop behavior of the process for which the model is developed. This does not have to be a problem, but it does need to be taken into account when developing a model from these data. These issues are a topic of future research and do not impose a problem here. For this case, virtually no process control is available (except for the input flow rate, which is considered a constant). Furthermore, the process is not stationary and it is assumed that sufficient experiments can be carried out to obtain representative data about the process behavior.

Since α is not measurable, it was estimated using an extended Kalman filter as described in [6,7]. The use of estimations introduces additional uncertainty. However, it is seen as the best way to obtain information about α .

In the filter, no additional information about the net growth rate α was assumed, except that it is a function of the substrate concentration S and the biomass concentration X . The filter was implemented in discrete form using the state Eqs. ((2)–(5)), supplemented by:

$$\alpha_{k+1} = \alpha_k + w_{\alpha,k} \quad (6)$$

where subscript k denotes the time step. Δt is the sampling time and $w_{\alpha,k}$ is the process noise for α at time step k . Process noise is simulated by adding white noise to the original process state (S , X , P and V) at every time step. This noisy state was then used to calculate the state at the next time step, so some correlation in the process noise occurs.

The process state vector for the Kalman filter is defined as $\mathbf{X}_{\text{Kalman},k} = \text{col}(\alpha_k, X_k, S_k, P_k, V_k)$. Using the model equations, the filter is able to make estimations of α based on both process measurements (generated by the reference model) and the estimates of X , S , P and V . The filter was tuned by adjusting the diagonal elements of the noise covariance matrix \mathbf{Q} using trial and error. The initial process state was taken the same as the process measurements at $t=0$. An initial estimate for the unmeasured α at $t=0$ was also made to initialize the filter. The filter results were analyzed by observability and robustness measures from [8] and by reviewing the autocorrelation of the innovation of the filter.

The Kalman filter that was developed estimates α for one batch run only. Although this results in input–output data in the form of estimates of S , X , and α , data from one batch run do not provide sufficient information for the input–output mapping for α . Figure 2a illustrates this. To obtain sufficient information about the mapping, the filter is applied to several batch runs, each executed under different conditions, by varying the input flow rate and the initial conditions of the reactor. This provides more information about the relationship between S , X , and α , as shown in Fig. 2b.

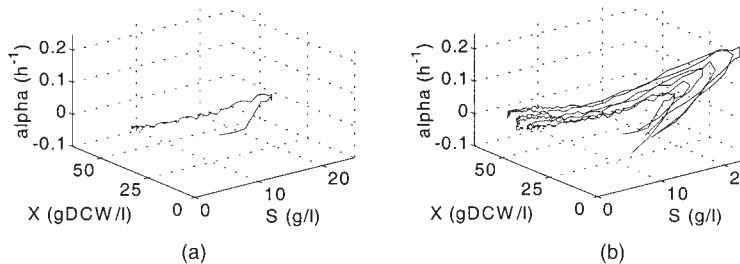


FIGURE 2 α as a function of S and X for one batch run (a); and α as a function of S and X for 6 additional batch runs (b).

7. IDENTIFICATION OF THE FUZZY SUBMODEL

With the identification data set available, the fuzzy submodel for the growth rate α can be developed. In this paper, three identification techniques will be compared, each of which is described in the following sections. The techniques will be used in such a way, that they can easily be compared. The resulting fuzzy models will make use of the same type of membership functions and will have similar complexity.

7.1. Fuzzy Clustering

The basic idea behind clustering is to divide a set of objects into self-similar groups (clusters), using the available data. Clustering methods usually are based on assumption about the geometry of the clusters that need to be determined, which include spheres, lines, hyperplanes, ellipsoids etc. Various clustering algorithms can be used to develop fuzzy models; a useful overview of different techniques can be found in [9]. A complete review of different clustering techniques is beyond the scope of this paper; we will focus on a well-known clustering technique for developing fuzzy models: Gustafson–Kessel (GK) clustering [10]. A detailed description of the clustering technique used in this paper is given in [11].

Using the GK algorithm, it is possible to fit an *a priori* specified number of hyperplanes through data, approximating the system under investigation by a collection of local linear models. The fuzzy system which is determined is therefore of the Takagi–Sugeno–Kang (TSK) type. The GK algorithm searches for ellipsoidal clusters. The clustering procedure is an iterative process, in which the cluster centers are moved in the input–output space until some convergence criterion has been met. The membership values for the data features are calculated based on the distance to the cluster centers: the closer a feature is to a cluster center, the higher its membership value for that cluster is. Results about the location and the shape of the clusters is given in the form of cluster centers v_i , cluster covariance matrices F_i and a fuzzy partition matrix U , containing the membership values of the data features for each of the clusters. The clustering algorithm is shown in Fig. 3.

Depending on the data that is clustered, it is not always possible to determine the correct number of clusters beforehand. A tool to help overcome this problem is the Modified Compatible Cluster Merging algorithm (MCCM) [11,12]. The main features of this technique are cri-

Initialise:

- cluster centers v_i randomly
- covariance matrices F_i with the identity matrix
- number of clusters k

While $\max |\Delta\mu_{ij}| < \varepsilon_{GK}$

- compute the distances of the data features to the cluster using

$$d^2(\mathbf{x}_j, P_i) = (\mathbf{x}_j - v_i)^T |F_i|^{1/n} F_i^{-1} (\mathbf{x}_j - v_i)$$

- compute the membership grades

$$\mu_{ij} = \frac{(d^2(\mathbf{x}_j, P_i))^{-1/m_e-1}}{\sum_{l=1}^k (d^2(\mathbf{x}_j, P_l))^{-1/m_e-1}}$$

if $d^2(\mathbf{x}_j, P_i) = 0$ for some $i=k$, then set $\mu_{ij} = 1$
and for all $i \neq k$ set $\mu_{ij} = 0$

- compute new cluster center

$$v_i = \frac{\sum_{j=1}^n \mu_{ij}^{m_e} \mathbf{x}_j}{\sum_{j=1}^n \mu_{ij}^{m_e}}$$

- compute new covariance matrix

$$F_i = \frac{\sum_{j=1}^n \mu_{ij}^{m_e} (\mathbf{x}_j - v_i)(\mathbf{x}_j - v_i)^T}{\sum_{j=1}^n \mu_{ij}^{m_e}}$$

FIGURE 3 GK-clustering algorithm.

teria which measure the level of compatibility of the clusters. These are based on the distances between clusters and their orientation with respect to each other. If two clusters are sufficiently close, or have the same orientation, they should be merged. A threshold value is used to determine whether clusters are close or have the same orientation. If clusters are to be merged, the GK algorithm is executed again, initialized with the new number of clusters, determined by the MCCM algorithm.

After clustering is finished, the fuzzy model can be constructed. Every cluster will be represented by one rule in the fuzzy model. The premise part of the fuzzy model is determined by projecting the fuzzy partition matrix U on the appropriate input axes. Since this matrix contains membership information about the clusters with respect to the features of the identification data set, the premise part can be constructed by fitting membership functions through these projections. In this work, double sigmoid membership functions are used:

$$\mu(x, a, b, c, d) = \frac{1}{1 + \exp(-a(x - b))} \cdot \frac{1}{1 + \exp(c(x - d - \delta))} \quad (11)$$

where $a, c \in \mathbb{R}^+$ determine the slope of the membership functions (their “fuzziness”) and $b \in \mathbb{R}$ and $d \in \mathbb{R}^+$ determine its position. δ is a parameter that makes sure that the maximum value of the membership function is always approximately 1. This way, the fuzzy set described by the membership function is always transparent (i.e. interpretable). δ is calculated as follows:

$$\delta = -\left(\frac{1}{a} + \frac{1}{c}\right) \ln\left(\frac{1}{0.99} - 1\right) + b \quad (12)$$

This equation can be derived from the demand that the upward branch of the double sigmoid (first term in Eq. (11)) reaches almost one (0.99) before the downward branch (second term in Eq. (11)) starts. Rewriting Eq. (11) yields the expression for δ . When the fuzzy partition matrix is projected, valuable information about the shape of the clusters can be lost. To overcome this problem, the clusters can be rotated to yield clearer projections. For a description of this

procedure, the reader is referred to [11]. An alternative to rotation is the use of multidimensional membership functions, omitting the need for projection [13]. Neither techniques were necessary here.

Finally, the consequence part of the fuzzy model can be calculated directly from the cluster covariance matrices F_i , since the eigenvectors of these matrices will give information about the orientation of the hyperplanes. These hyperplanes will be described by the consequence part of the fuzzy model. Calculation of the consequence part parameters can be done using the $n - 1$ eigenvectors corresponding with the $n - 1$ largest eigenvalues of a cluster, which span the hyperplane [11], or with the eigenvector corresponding to the smallest eigenvalue, which determines the normal vector to the hyperplane [14]. Least squares methods can also be used to determine the consequence part parameters [14].

Summarizing, the complete identification algorithm is as follows:

1. Choose the process input variables and output variable. Determine the initial number of clusters k_{\max} .
2. Perform GK-clustering.
3. Perform MCCM cluster merging. If clusters are to be merged, go to step 2 and recluster, using the number of clusters determined by the merging algorithm. If not, go to step 4.
4. Determine the premise part of the fuzzy model by projection of the fuzzy partition matrix U and rotation, if necessary, or by use of multidimensional membership functions.
5. Determine the consequence part of the fuzzy model from the cluster covariance matrices F_i .

7.2. Genetic Algorithms

Genetic Algorithms (GAs) are well known for their optimization capabilities. Following basic Darwinistic propagation, the method is based on a “survival of the fittest” principle, where only the solution candidates with the best desirable properties (e.g. smallest error) will survive. In essence, a “population” of possible solution candidates for the optimization problem (e.g. models that need to describe

measurement data as good as possible) is created in an “environment”, in which only a limited number of candidates can survive. The candidates that will survive are selected by evaluating their fitness value through the fitness function (similar to the objective function in more traditional optimization algorithms). The candidates with the lowest fitness value are removed, the candidates with the highest fitness values are propagated into the next “generation” (iteration step). During this propagation, “reproduction” occurs, in which the best candidates are copied to take the place of the candidates that are removed from the population. Some random effects can occur, such as “mutation” (random change in the properties of a candidate solution) and “cross over” (exchange of information between two candidate solutions). The procedure is iterative and is terminated when some convergence criterion is met, usually when the maximum number of iterations is reached or when there is no more increase in fitness.

The possible solution candidates are coded in the form of parameter strings, usually utilizing binary coding. These codings are often called “chromosomes”. In GAs, reproduction, mutation and crossover are all executed on these chromosomes.

Compared to conventional optimization algorithms, GAs are different in a number of ways [15]:

- GAs work with a coding of the parameter set, not the parameters themselves
- GAs search from a population of points, not a single point, and therefore are more robust with respect to becoming trapped in local optima than other algorithms
- GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge
- GAs use probabilistic transition rules, not deterministic rules, i.e., GAs use random experiments to select new candidates which are *likely* to be better.

A useful overview of theory and applications of GAs can be found in [15].

Using GAs, it is possible to develop a fuzzy system and many applications have already been reported [16]. Using GAs to optimize fuzzy systems, the fuzzy system needs to be coded into chromosomes. There are numerous texts available which describe different approaches to the coding problem. Here, a relatively straightforward coding scheme is used. A distinction is made between coding of linguistic (Mamdani) type fuzzy models and TSK type fuzzy models.

The objective function (or fitness function) that was used is the same for both types of fuzzy models, and is given by:

$$J(\varphi_{\text{fuzzy}}) = \left(\frac{1}{M} \sum_{i=1}^M (f_{\text{fuzzy}}(\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,n-1}) - \mathbf{x}_{i,n})^2 \right)^{1/2} \quad (13)$$

which represents the root mean squared error with respect to the n dimensional identification data \mathbf{x} .

When coding linguistic fuzzy models, the parameters of all of the membership functions are coded into the chromosome, so that the membership functions of the fuzzy model can be optimized. In addition, the rule structure can also be coded, enabling rule structure optimization. Figure 4 shows how the chromosomes are constructed. The model parameters are supplied to the GA using real-valued (non-binary) parameters. The GA translates these real values to a binary representation. Therefore, extra information needs to be supplied to the GA about the parameters, such as an allowed parameter interval (to which the binary coding is mapped) and a parameter

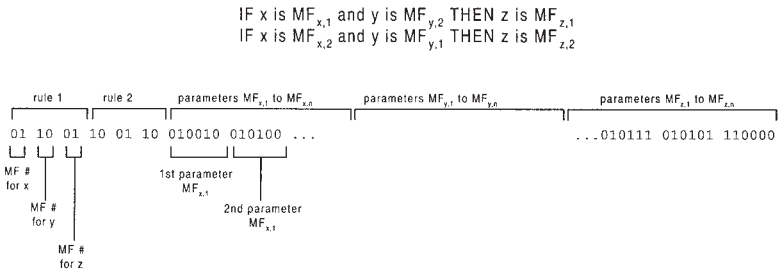


FIGURE 4 Coding of a 2-rule linguistic fuzzy model into a chromosome.

resolution (to determine the bit string length per parameter). These can be chosen individually per parameter.

The same coding scheme can be followed when coding TSK type fuzzy models. However, as mentioned before, when the premise part of the fuzzy model is known, the consequence part can be calculated easily using least squares methods. Therefore, only the premise part parameters and the rule structure are coded into chromosomes, and the consequence part parameters are calculated directly using a least squares approach, without intervention of the GA. This is similar to the learning algorithm that is used in the Adaptive Network based Fuzzy Inference system (ANFIS) method [17]. Coding TSK models this way reduces the length of the chromosomes (the last part can be omitted, see Fig. 4) and the search space of the optimization algorithm, improving the speed and performance of the GA.

The consequence part is calculated as follows [14]. The consequence part of the rule i of the fuzzy model for some n dimensional data vector \mathbf{x} with $n - 1$ inputs is given by:

$$y_{f,i} = [\mathbf{x}_i \quad \dots \quad \mathbf{x}_{n-1} \quad 1] \boldsymbol{\theta}_i \quad (14)$$

in which $y_{f,i}$ is the output of rule i and $\boldsymbol{\theta}_i$ is the consequence part parameter vector for rule i .

Given the identification data set, consisting of M n -dimensional vectors \mathbf{x} , define a M by n matrix \mathbf{P} , with the first $n - 1$ elements of \mathbf{x}_k on the k th row. Append this matrix with a unitary column as the n th column. Furthermore, define the vector \mathbf{y} with the n th element of \mathbf{x}_k on the k th row. This gives:

$$\mathbf{P} = \begin{bmatrix} \mathbf{x}_{1,1} & \dots & \mathbf{x}_{1,n-1} & 1 \\ \mathbf{x}_{2,1} & \dots & \mathbf{x}_{2,n-1} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{x}_{M,1} & \dots & \mathbf{x}_{M,n-1} & 1 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{x}_{1,n} \\ \mathbf{x}_{2,n} \\ \vdots \\ \mathbf{x}_{M,n} \end{bmatrix} \quad (15)$$

To calculate the consequence part parameters vector $\boldsymbol{\theta}_i$ of rule i of the fuzzy model, first multiply each row k of \mathbf{P} and \mathbf{y} with the square root of

the degree of fire of rule i for data vector k , $\beta_{i,k}$. This gives:

$$\tilde{\mathbf{P}}_i = \begin{bmatrix} (\beta_{i,1})^{1/2} \mathbf{x}_{1,1} & \dots & (\beta_{i,1})^{1/2} \mathbf{x}_{1,n-1} & (\beta_{i,1})^{1/2} \\ (\beta_{i,2})^{1/2} \mathbf{x}_{2,1} & \dots & (\beta_{i,2})^{1/2} \mathbf{x}_{2,n-1} & (\beta_{i,2})^{1/2} \\ \vdots & \ddots & \vdots & \vdots \\ (\beta_{i,M})^{1/2} \mathbf{x}_{M,1} & \dots & (\beta_{i,M})^{1/2} \mathbf{x}_{M,n-1} & (\beta_{i,M})^{1/2} \end{bmatrix}, \quad (16)$$

$$\tilde{\mathbf{y}} = \begin{bmatrix} (\beta_{i,1})^{1/2} \mathbf{x}_{1,n} \\ (\beta_{i,2})^{1/2} \mathbf{x}_{2,n} \\ \vdots \\ (\beta_{i,M})^{1/2} \mathbf{x}_{M,n} \end{bmatrix}$$

The solution of the least squares problem of $\mathbf{y} = \mathbf{P}\boldsymbol{\theta}_i + \varepsilon$, is given by:

$$\boldsymbol{\theta}_i = [\tilde{\mathbf{P}}_i^T \tilde{\mathbf{P}}_i]^{-1} \tilde{\mathbf{P}}_i^T \tilde{\mathbf{y}}_i \quad (17)$$

This procedure is repeated for each rule in the model.

With TSK models, as with linguistic fuzzy models, it is possible to code only the membership function parameters or both the parameters and the rule structure. However, rule structure optimization may not always be necessary. For fully connected rule bases, the optimization structure proposed here does not need rule base optimization. Since every possible combination of antecedent membership functions is present, and since the calculation of the consequence parameters is based on the premise part of the rule structure, calculation of the consequence parameters will yield the same results for the situation with and without rule base optimization. The only difference will be that the rules will be in a different order. For cluster-like connected rule bases, every antecedent membership function occurs only once. This way, the rule base described independent “clusters” in the input space. Since these clusters can be moved around freely by the GA, rule structure optimization by the GA is not necessary.

The membership functions used in the antecedent part are double sigmoids, the same as used with clustering. The optimization procedure can be refined (constrained) by imposing demands on the level of overlap of the antecedent membership functions and their location, for

example. Although this imposes restrictions on the optimization, it does ensure that the resulting fuzzy models remain transparent, which in this case is seen as a requirement. The constraints were implemented for each input variable in the following way:

$$\left. \begin{array}{l} a_i = c_{i-1} \\ b_i = d_{i-1} + \delta \end{array} \right\} \quad i = 2, \dots, n_{MF} - 1 \quad (18)$$

with $a_i \dots d_i$ denoting the membership function parameters as in Eq. (11) for membership function i and n_{MF} the number of membership functions on the concerning input. The first and the last membership functions on the input variable are chosen as shouldered membership functions, described by single sigmoids. The restrictions make sure that two adjacent membership functions sum up to one: they form a fuzzy partition. In addition to preservation of transparency, the number of parameters for optimization is reduced. These constraints are not necessary for cluster-like connected rule bases, because for these rule bases the locations of the described “clusters” do not have to be orthogonal, as is the case with fully connected rule bases.

To summarize, only the parameters of the antecedent membership functions are coded into chromosomes. These chromosomes are thus a special case of the general representation illustrated in Fig. 4.

7.3. Neuro-Fuzzy Methods

Neuro-fuzzy methods can be viewed upon as a combination of fuzzy systems and artificial neural networks (ANNs). The fuzzy inference system is implemented in the framework of these adaptive networks. This provides the possibility to use backpropagation learning rules, commonly used to train these nets. Several approaches have been developed in the past. Some of these have been applied within the context of hybrid modeling, such as NEFPROX [18] and ASMOD [19], covering both linguistic and TSK type fuzzy models. For TSK models, Jang’s Adaptive Network based Fuzzy Inference System (ANFIS) approach is used [17].

Jang interprets a TSK type fuzzy system as an adaptive network, on which adaptive learning rules can be applied to optimize the system

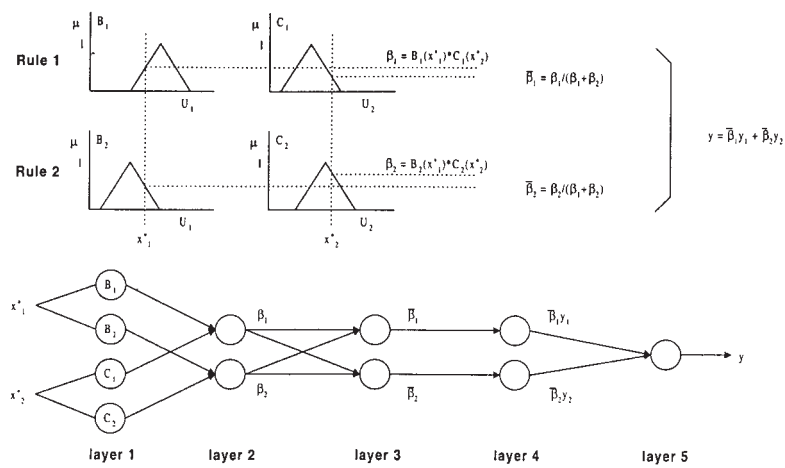


FIGURE 5 2 by 1 TSK model and corresponding ANFIS interpretation; the nodes in layer 1 contain the membership functions, layer 2 determines the degree of firing, layer 3 calculates the weighted degree of firing, layer 4 calculates the weighted rule outputs and layer 5 calculates the crisp model output.

parameters. The interpretation is illustrated in Fig. 5. Parameter optimization is carried out using a so-called *forward pass* and a *backward pass*. During every iteration, the consequence part parameters of the fuzzy model are determined using the same least squares approach as used with GA's (see Eqs. ((14)–(17))). An alternate approach to the least squares calculation given by Eqs. ((14)–(17)) is an iterative calculation of the solution to the least squares problem [17]. In this approach, no matrix inversion is used which results in a computationally more efficient calculation. However, for the problem presented here and the software available today, matrix inversion is not a problem. Calculation of the consequence parameters is called the forward pass. In the backward pass, the error rates propagate backwards and the premise part parameters are updated by a standard gradient descent learning rule (see [17]). These steps are performed iteratively until some criterion is satisfied, for example, until the change in parameters is very small.

The procedure starts with an initial fuzzy model. This model is trained by optimizing its membership function parameters, while the rule structure is left intact. This means that the rule structure needs

to be defined beforehand. Determining the rule base can be done by using heuristic knowledge or visual inspection of the input–output data. An initial rule base can also be constructed by performing clustering in the input space of the data. This is not investigated here. The double sigmoid membership functions described earlier are also deployed here.

8. HYBRID MODELING RESULTS

8.1. Data Preparation

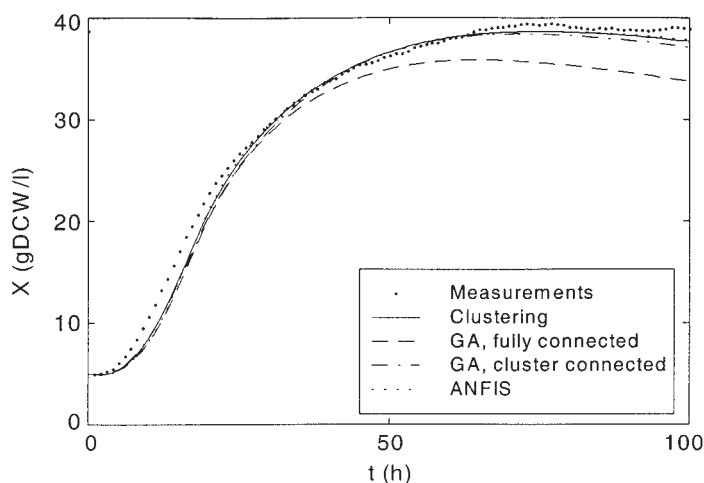
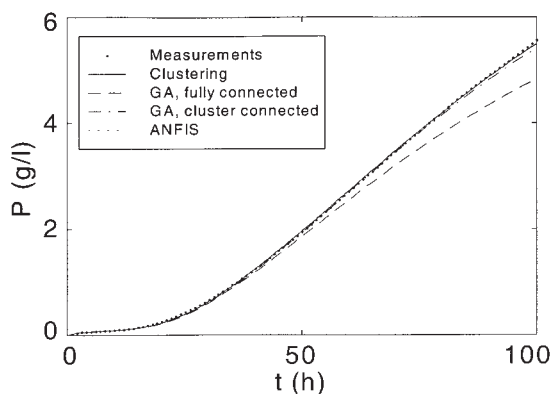
Because of the quadratic optimization criteria, the distribution of the data features is important for all of the identification techniques used in this research. The data set that was generated by the Kalman filter shows areas with high and low density of data features, so the set was manipulated to obtain a more even distribution of the data features in the input space. This was done by performing a heuristic step: all features present in the data set must have some minimum distance to its neighbors. The data set, generated by 16 batch runs under different conditions, contains 129 data features after this step.

8.2. Fuzzy Clustering

The identification data set was supplied to the clustering algorithm. The initial number of clusters was set relatively high, and the cluster merging threshold was set using trail and error. Settings for the clustering algorithm are shown in Table II. The resulting fuzzy submodel for α as a function of S and X consists of three rules (see appendix B, Fig. 13). Hybrid modeling results are shown in Figs. 6–8. The figures show reference measurements of X and P of a batch run, as well as the results of a free run of the hybrid model. That is, the model is initialized with a given state and the model uses its own state calculations to simulate the process. Thus modeling errors are propagated as the

TABLE II Settings clusteralgorithm

Initial number of clusters k_{\max}	5
Cluster merging threshold γ	0.5
Cluster termination criterion ε_{GK}	0.01

FIGURE 6 Hybrid modeling results: X for one batch run.FIGURE 7 Hybrid modeling results: P for one batch run.

simulation advances. If the model is used for, say, one-step-ahead prediction using measurement feedback, these discrepancies would not occur. However, one of the intentions of the model was that it could be used for off-line simulation studies, and this is why it was decided to present free runs. Furthermore, the corresponding trajectories of α are given. A more elaborate discussion will follow later. Model errors are shown in Table VI.

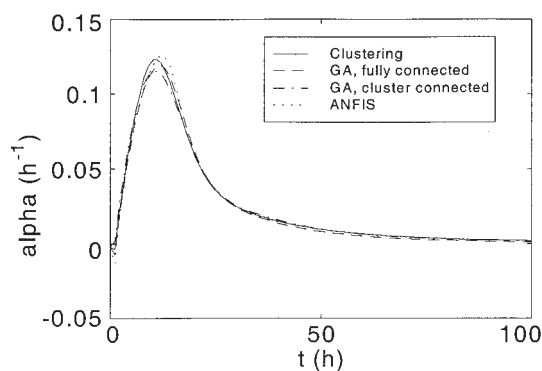
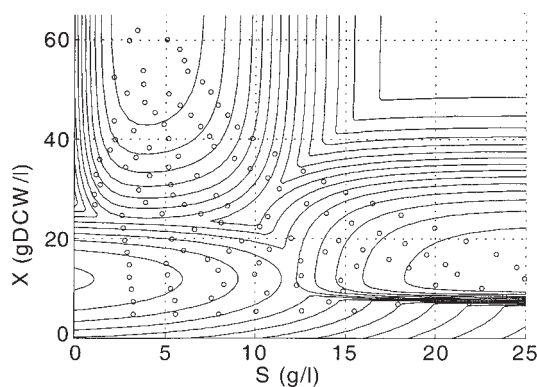
FIGURE 8 Hybrid modeling results: α for one batch run.

FIGURE 9 Input space and rule locations for fuzzy model obtained with clustering. Contour plot indicates degree of fire, dots indicate input data.

The three rules can be interpreted physically as each describing an operation phase of the batch reactor (see Fig. 9). The first rule describes α in the first phase of the batch run (low X , relatively high S), the third rule describes α at the end of a batch run under normal conditions (high X , low S) and the second rule describes an intermediate phase.

8.3. Genetic Algorithm

A comparison is made between a fully connected rule base and a rule base with cluster-like connections. The initial model contains four

rules for both type of connections. Initialization of the premise part membership functions does not play a role here: this is done by the GA, randomly. Settings of the GA are shown in Tables III and IV, the resulting fuzzy models can be found in appendix B (Figs. 14 and 15).

For the fuzzy model with the fully connected rule base, the optimization resulted in maximum values for the a and c parameters of all the membership functions. This is a result of the interpolation properties of TSK models and has been discussed before [11,14]. These parameters determine the level of overlap between the fuzzy sets, i.e. the “fuzziness” of the sets. The higher these parameters, the crisper the sets. To prevent the sets from becoming too crisp, suitable maximum values for a and c were determined experimentally. Hybrid modeling results for a free run of the model are shown in Figs. 6–8.

For the cluster-like connected rule base, the GA had the tendency to move all of the clusters outside the input space (Fig. 10). A suitable fit could still be found, because the sigmoid membership functions never become zero, so the fuzzy inference mechanism still functions properly.

TABLE III Settings GA, fully connected rule base

<i>TSK model, fully connected rule base</i>	
Number of membership functions on S	2
Number of membership functions on X	2
Number of rules	4
Number of chromosomes	77
Number of generations	77
Propagation selection criterion	Tournament
Probability of crossover (1 point crossover)	0.77
Probability of mutation	0.0077

TABLE IV Settings GA, cluster-like connected rule base

<i>TSK model, cluster-like connected rule base</i>	
Number of membership functions on S	2
Number of membership functions on X	2
Number of rules	2
Number of chromosomes	77
Number of generations	77
Propagation selection criterion	Tournament
Probability of crossover (1 point crossover)	0.77
Probability of mutation	0.0077

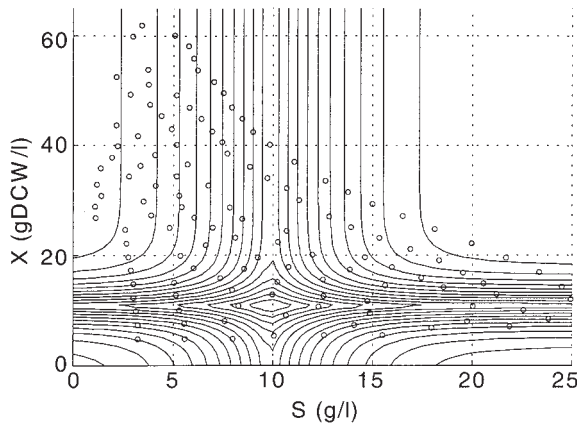


FIGURE 10 Input space and rule locations for fuzzy model obtained with genetic algorithm, fully connected rule base. Contour plot indicates degree of fire, dots indicate input data.

Extra penalties were introduced in the objective function to overcome this: fuzzy models with clusters that describe areas without identification data in the input space are reduced in fitness. This resulted in a suitable model. Initially, a run with four clusters was made. However, this resulted in a model where some of the clusters completely overlapped, indicating a too complex model. Eventually, a model with 2 clusters was developed (see appendix B), resulting in a response shown in Figs. 6–8.

Physically, the interpretation of the rules is similar to the interpretation given for the fuzzy model that was constructed with fuzzy clustering (Fig. 11). It can be seen that the rules make a distinction between the first phase and the latter phase of a batch run under normal conditions (i.e., without an extraordinary high biomass concentration at $t=0$). This is consistent with what would be expected, because most of the dynamics in α are present in the first phase of a batch run.

8.4. Neuro Fuzzy Method

As with GA, optimization of a four rule fuzzy model with fully connected rule base was compared with optimization of a four rule fuzzy model with a cluster-like connected rule base. However,

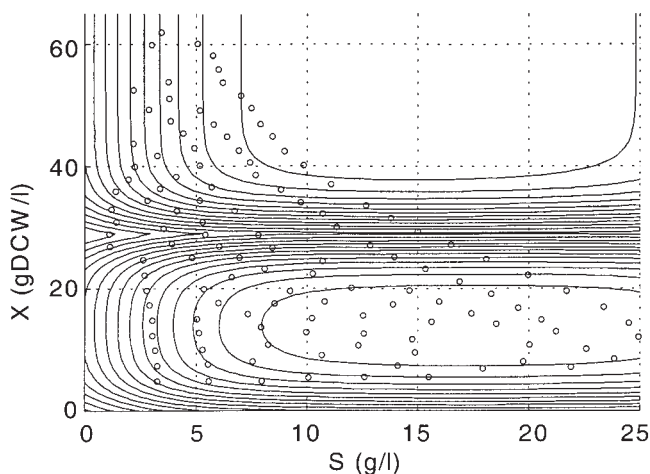


FIGURE 11 Input space and rule locations for fuzzy model obtained with genetic algorithm, cluster-like connected rule base. Contour plot indicates degree of fire, dots indicate input data.

initialization plays an important role. If both type of models are initialized so that they are essentially the same (same degree of fire of the four rules and same consequence parameters), it was found that optimization leads to virtually the same results. In that case, a fully connected rule base is preferable, because this reduces the number of model parameters.

Careful inspection of the data set can be used to determine a more suitable premise part of the initial fuzzy model instead of a general initialization. However, this may be cumbersome and not always possible with high dimensional input spaces. Inspection of the data set used here resulted in a simple 4-rule model with a fully connected rule base and two evenly spaced sigmoid membership functions on both S and X .

Settings for ANFIS are shown in Table V. The fuzzy model identified with ANFIS can be found in appendix B, Fig. 16. Results of the hybrid model are shown in Figs. 6–8. As can be seen, the interpretability with respect to X is deteriorated by the optimization. The overlap between the two sets has become large and the membership functions do not reach 1. Furthermore, the output of rule four is out of proportion with respect to the other three rules. This is the result of

TABLE V Settings ANFIS

<i>TSK model, fully connected rule base</i>	
Number of membership functions on S	2
Number of membership functions on X	2
Number of rules	4
Number of training epochs	1000
Initial learning rate	0.01
Learning rate decrease rate	0.9
Learning rate increase rate	1.1

optimization in an area where no data is available. This shows that extrapolating this fuzzy model has indeed no physical meaning.

The effect of the optimization of the consequence part is much larger than the effect of the optimization of the premise part. If only the consequence parameters of the initial fuzzy model, before premise part optimization, are optimized using the least squares method, the RMSE will drop to the same order of magnitude as the RMSE after premise part optimization. Therefore, changes in the form of premise part membership function parameter adjustments will have a relatively small effect on model performance. This could be an explanation for the deterioration of the interpretability of the premise part of the fuzzy model: the shape of the membership functions is less relevant because changes in shape and position will not have a large effect on the performance criterion. The reason why this does not occur with the GA is because the GA searches the optimization space much more effectively. Instead of following a steepest descent path (leading to a local optimum), the GA optimizes several possible solutions distributed randomly in the search spaces and is thus less likely to get stuck in one optimum.

The physical interpretation of this fuzzy model is much more complicated, because the adjustments of the premise part are dependent on the initialization and, in this case, are minor (see Fig. 12).

9. COMPARISON

Since all of the fuzzy submodels use the same fuzzy model structure and the same type of membership functions, comparison of the per-

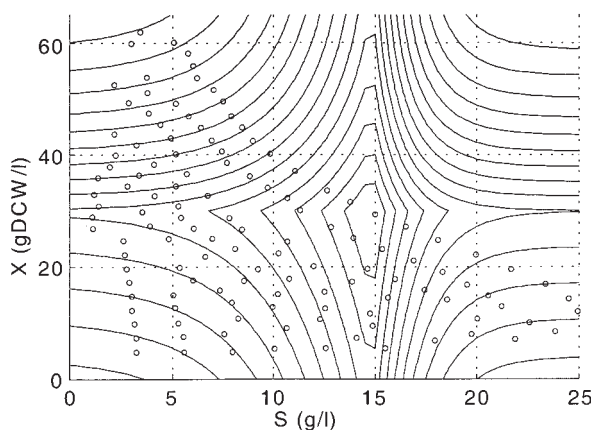


FIGURE 12 Input space and rule locations for fuzzy model obtained with ANFIS. Contour plot indicates degree of fire, dots indicate input data.

formance of the different technique is possible. The most important criteria for comparison in this work are:

- Modeling errors.
- Transparency and interpretability of the identified models.
- Sensitivity to initialization.

With respect to modeling errors, it can be seen in Figs. 6–8 that all of the constructed hybrid models perform comparably. All techniques are capable of producing a fuzzy submodel with acceptable performance. The differences between the hybrid models are caused by the various submodels for α , since the physical model structure is the same for all of the hybrid models. This discrepancy with the measurements is caused by a series of errors. First of all, the fuzzy submodels are identified using *estimates* of α . Furthermore, the submodels are fit to these estimates, introducing fitting errors. Finally, as mentioned before, the runs in Figs. 6–8 are free runs, which means that model errors are propagated through the simulation run by integration and will increase in magnitude. Modeling errors can be reduced by optimizing the parameters of the fuzzy submodel once the hybrid model is constructed, by using hybrid model output and state measurements (Table VI). This is a topic for future research. Since every identification technique yielded a suitable fuzzy submodel, it is interesting to make a

TABLE VI Hybrid modelling errors

<i>Identification method</i>	$\varepsilon_{\text{RMSE}}$	$\varepsilon_{\text{integrated, X}}$
Fuzzy Clustering	0.0083	72.82
Genetic Algorithm, fully connected	0.0083	249.9
Genetic Algorithm, cluster-like connected	0.0089	93.15
Neurofuzzy method	0.0071	70.54

comparison about the construction of the fuzzy submodels using the different techniques rather than a comparison about model performance.

The main advantage of fuzzy clustering is that it automatically determines the fuzzy model structure. Initialization was not an issue. With respect to transparency and interpretability, the clustering algorithm performed very well. The model that was obtained is simple and it was possible to give a physical interpretation afterwards.

Although computationally demanding, the two-step identification approach of the genetic algorithm worked especially well with cluster-like connected rule bases. This gave the GA the possibility to shift rules independently of each other. It was also possible to give a physical interpretation of the model. However, the fuzzy model structure must be given in advance. This is a disadvantage: it may be very difficult to define an optimal fuzzy model structure in advance. Some researchers have proposed to use clustering in the input space of the system to obtain an initial idea about the fuzzy sets that are required, but for the functional relationships investigated in this work clustering in the input–output space seems more logical. An advantage of the GA is that it can be used to “train” a complete hybrid model. Since the GA works with an initial fuzzy submodel, this submodel can be used to construct an “initial hybrid model”. Thus instead of using input–output data of the uncertain relation to train the fuzzy submodel, the GA can also use output data of the hybrid model to train the fuzzy submodel within the hybrid model. In this case, the need for the Kalman filter is eliminated. The proposed approach is currently under investigation.

A similar approach can be followed using a neuro-fuzzy identification method. However, since initialization of the fuzzy submodel before optimization has a high impact on the result, this may be a difficult task. Although a good fuzzy submodel was obtained using ANFIS, the method had the tendency to change the membership

functions in such a way that interpretation was often difficult, while the location of the membership functions was changed very little. This makes the method very sensitive to initialization. Consequently, the other two identification methods are more desirable alternatives.

10. CONCLUSIONS

In this paper, we presented a successful approach to constructing hybrid fuzzy-first principles models from process data. Three different identification techniques for the construction of the fuzzy submodel for the net growth rate α were applied successfully and compared. Of these three, fuzzy clustering was found to be the best alternative, because it determines the fuzzy model structure itself from the process data at hand. The cluster-like connected rule base results in a transparent and interpretable fuzzy submodel with excellent simulation performance.

A genetic algorithm approach is a good alternative to fuzzy clustering, with cluster-like connected rule bases giving good results. Although the model is less interpretable than with fuzzy clustering, the approach has the advantage that it can be applied so that no input-output data or estimations of the uncertain relation need to be available, which is an advantage above fuzzy clustering. However, in this case, a deliberation needs to be made between the excellent transparency of fuzzy models constructed with clustering (but additional uncertainty introduced by estimations) and the somewhat lesser interpretability of fuzzy submodels constructed with a GA (but with no additional uncertainty caused by estimations).

The major drawback of ANFIS is its sensitivity to initialization of the problem. As with GA, the structure must be provided, as an initial guess of membership parameters. An optimum is found from there. Other neuro-fuzzy approaches, like ASMOD and NEFPROX, provide mechanisms of determining the model structure automatically. These mechanisms could be incorporated in the ANFIS technique. Given the requirements for this research, genetic algorithms and fuzzy clustering provide good alternatives, with fuzzy clustering the best option, if input-output data or estimations are available.

Acknowledgements

The contributions of Coen Noppert and Yücel Kök are gratefully acknowledged.

NOTATION

c_L	Cell lysis rate (h^{-1})
c_{Lm}	Constant (h^{-1})
D	Dilution rate, F/V (h^{-1})
F	Feed flow rate (l/h)
f_{fuzzy}	Fuzzy model
F_i	Covariance matrix for cluster i
J	Objective function for the GA
K	Product decay constant (h^{-1})
k	Number of clusters
K_I	Constant (g/l)
K_L	Constant (g/l)
K_P	Constant (g/l)
K_X	Constant (—)
M	Number of identification data features
m_e	Fuzzy exponent
m_x	Maintenance energy factor
m_{xm}	Constant (h^{-1})
n	Dimension of the input–output space of the identification problem
P	Product concentration (g/l)
q_p	Product formation rate (h^{-1})
q_{qm}	Constant (h^{-1})
S	Substrate concentration (g/l)
S_f	Substrate concentration in the feed (g/l)
V	Reactor volume (l)
\mathbf{w}_k	Process noise vector at time step k
X	Biomass concentration (gDCW/l)
\mathbf{x}	n -dimensional data feature vector
$\mathbf{x}_{\text{Kalman},k}$	Process state vector for the Kalman filter
$Y_{P/S}$	Yield factor which relates production rate and substrate consumption

$Y_{X/S}$	Yield factor which relates cell growth rate and substrate consumption
α	Net growth rate $\mu - c_L$ (h^{-1})
$\beta_{i,k}$	Degree of Fire (DOF) of rule i for data feature k
δ	Additional parameter for double sigmoid membership function (—)
$\varepsilon_{\text{integrated}}$	Integral error
ε_{GK}	Termination criterion for the GK algorithm
$\varepsilon_{\text{RMSE}}$	Root Mean Squared Error
θ_i	Fuzzy model consequence part parameters for rule i
μ	Specific growth rate (h^{-1})
μ	Membership value (—)
μ_m	Constant (h^{-1})
v_i	center of cluster i
σ	Substrate consumption rate (h^{-1})
φ_{fuzzy}	Fuzzy model parameter vector

APPENDIX A: BIOREACTOR MODEL

The model describes four states of the process, namely the cell biomass concentration X (in gDCW/l, where DCW means dry cell weight), the substrate concentration S (in g/l), product concentration P (g/l) and the overall volume V (l). Four state equations describe these states

$$\frac{dX}{dt} = X(\mu - D - c_L) \quad (19)$$

$$\frac{dS}{dt} = -\sigma X + (S_f - S)D \quad (20)$$

$$\frac{dP}{dt} = q_p X - P(D + K) \quad (21)$$

$$\frac{dV}{dt} = F \quad (22)$$

with

S_f Substrate concentration in feed (g/l)

K product decay constant (h^{-1})

D dilution rate F/V (h^{-1})

All the other symbols govern specific rates which play a role in the process as defined below.

Growth rate

The equation for the specific growth rate μ is an adaptation of the Contois Eq. [20], which in itself is a special form of the well-known Monod equation.

$$\mu = \frac{\mu_m S}{K_X X + 10} \quad (23)$$

Cell lysis rate

Cell lysis, the dying of the biomass, is usually expressed by an exponential decrease in biomass. In this case, expressed by

$$c_L = \frac{c_{Lm} X}{K_L + X + 1} \exp(-S/100) \quad (24)$$

(Numbers in the equation may have units)

Product formation rate

The specific product formation rate q_p will vary with S and X according to

$$q_p = \frac{1.5 q_{pm} S X}{4 K_p + X S (1 + S/3 K_I)} \quad (25)$$

(Numbers in the equation may have units)

Substrate consumption rate

The rate of uptake of substrate by micro-organisms is generally considered to be either related to that of growth or to that required for cell maintenance. The specific substrate consumption rate is given by

$$\sigma = \frac{\mu}{Y_{x/s}} + \frac{q_p}{Y_{p/s}} + m_x \quad (26)$$

Maintenance energy

The substrate consumption rate needed for non-growth functions, the maintenance energy factor, is expressed by

$$m_x = \frac{m_{xm}X}{X + 10} \quad (27)$$

(Numbers in the equation may have units)

APPENDIX B: FUZZY MODELLING RESULTS

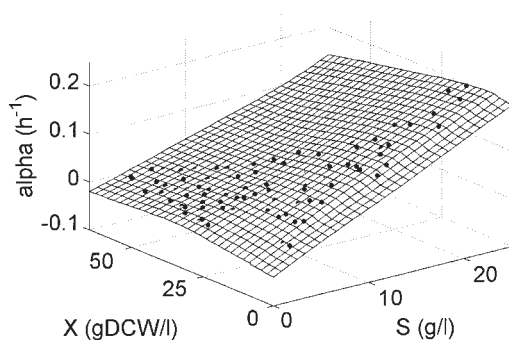


FIGURE 13 Fuzzy model for α identified with fuzzy clustering. Dots indicate identification data, the plane indicates the fuzzy model output.

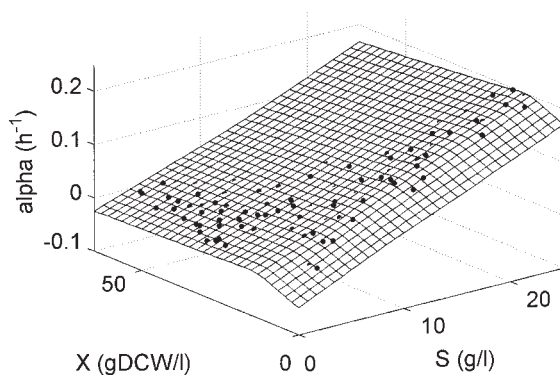


FIGURE 14 Fuzzy model for α identified with the GA, fully connected rule base. Dots indicate identification data, the plane indicates the fuzzy model output.

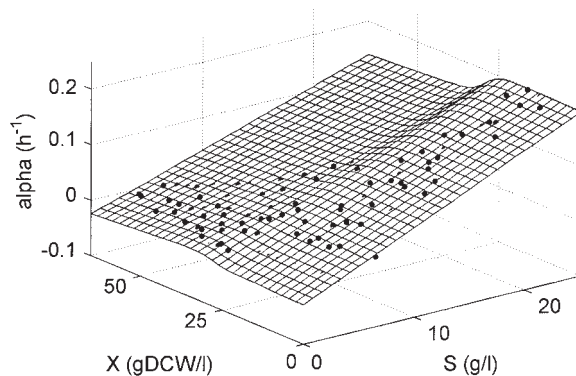


FIGURE 15 Fuzzy model for α identified with the GA, cluster-like connected rule base. Dots indicate identification data, the plane indicates the fuzzy model output.

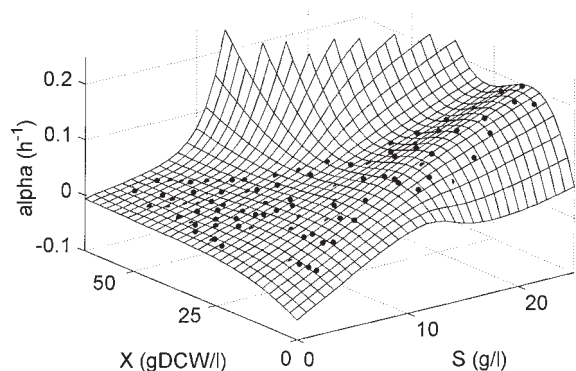


FIGURE 16 Fuzzy model for α identified with ANFIS. Dots indicate identification data, the plane indicates the fuzzy model output.

References

- [1] M.L. Thompson and M.A. Kramer (1994). Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal*, **40**(8), 1328–1340.
- [2] D.C. Psychogios and L.H. Ungar (1992). A hybride neural network-first principles approach to process modeling. *AIChE Journal*, **38**(10), 1499–1511.
- [3] O. Otto and K. Hartmann (1996). Hybrides Modell eines Sichters auf der Basis künstlicher Neuronaler Netze. *Chemie Ingenieur Technik*, **68**(12), 1578–1581.
- [4] J.A. Roubos, P. Krabben, M. Setnes, R. Babuska, J.J. Heijnen and H.B. Verbruggen (8–9 September 1999). *Hybrid model development for fed-batch bioprocesses; combining physical equations with the metabolic network and black-box kinetics*, in *6th UK Workshop on Fuzzy Systems*. pp. 231–239. Uxbridge, UK.

- [5] F.E. Cellier, A. Nebot, F. Mugica and A. De Alborno (1996). Combined qualitative/quantitative simulation models of continuous-time processes using fuzzy inductive reasoning techniques. *International Journal of General Systems*, **24**(1–2), 95–116.
- [6] B. Roffel and P. Chin (1987). *Computer control in the process industries*. Lewis Publishers, Chelsea, Michigan.
- [7] R.G. Brown and P.Y.C. Hwang (1992). *Introduction to random signals and applied Kalman filtering*. 2nd Ed., John Wiley and Sons, New York.
- [8] W.F. Ramirez (1994). *Process control and identification*. Academic Press, San Diego.
- [9] N.R. Pal, K. Pal, J.C. Bezdek and T.A. Runkler (1997). Some issues in system identification using clustering. *Oral presentation on the 1997 IEEE international conference on neural networks*. Houston, Texas, USA.
- [10] D.E. Gustafson and W.C. Kessel (1978). Fuzzy clustering with a fuzzy covariance matrix. *IEEE conference on decision and control*. pp. 761–766. January 1979, San Diego, Ca.
- [11] H.A.E. De Bruin and B. Roffel (1996). A new identification method for fuzzy linear models of nonlinear dynamic systems. *Journal of Process Control*, **6**(5), 227–293.
- [12] U. Kaymak and R. Babuska (1995). Compatible cluster merging for fuzzy modeling. in *FUZZ-IEEE/IFES'95*. pp. 20–24. March 1995, Yokohama, Japan.
- [13] P.F. Van Lith (1997). *Issues in hybrid fuzzy-first principles modeling*, in *Chemical Technology*. University of Twente: Enschede (NL).
- [14] R. Babuska (1996). *Fuzzy modeling and identification*, in *Electrical Engineering*. Technical University of Delft: Delft, The Netherlands.
- [15] L. Boullart (1997). Genetic algorithms: theory and applications. *Journal A Bruxelles EDS*, **38**(2), 13–23.
- [16] T. Back and F. Kursawe (1995). Evolutionary algorithms for fuzzy logic: a brief overview, In: B. Bouchon-Meunier, R.R. Yager and L.A. Zadeh (Eds.). *Advances in fuzzy systems*, volume 4, pp. 3–10. World Scientific, Singapore.
- [17] J.-S.R. Jang (1993). ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, **23**(3), 665–685.
- [18] D. Nauck and R. Kruse (1997). Function Approximation by NEFPROX in *Second European Workshop on Fuzzy Decision Analysis and Neural Networks for Management, Planning and Optimization*. 21–22 May 1996, Dortmund, Germany.
- [19] T. Kavli and E. Weyer (1999). On ASMOD – An Algorithm for Empirical Modelling using Spline Functions. SINTEF Instrumentation, Norway.
- [20] I.J. Dunn, J. Heinzle, J. Ingham and J.E. Prenosil (1992). *Biological reaction engineering, principles, applications and modelling with PC simulation*. VCH Verlagsgesellschaft, Weinheim, Germany.